

**A**

**PROJECT REPORT**

**FOR**

**SUBJECT: PROJECT PHASE II**

**ON**

**TASK MANAGEMENT WEB APPLICATION**

**FOR SOFTWARE PROJECTS**

Submitted in partial fulfilment of the requirement for the award of

**Bachelor of Technology**

**In**

**Computer Science and Engineering**

**Walchand Institute of Technology**

**(An Autonomous Institute)**

By

|  |  |
| --- | --- |
| **Piyush Agarwal** | **13** |
| **Venkat Dandi** | **14** |
| **Mahadev Dharmasale** | **15** |
| **Akchat Jain** | **16** |

Under the Guidance of

**Dr. Pratibha Kaladeep**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**WALCHAND INSTITUTE OF TECHNOLOGY**

**SOLAPUR - 413006**

**(2024-2025)**



**Certificate**

This is to certify that the project entitled

**TASK MANAGEMENT WEB APPLICATION**

**FOR SOFTWARE PROJECTS**

Is submitted by

|  |  |
| --- | --- |
| **Piyush Agarwal** | **13** |
| **Venkat Dandi** | **14** |
| **Mahadev Dharmasale** | **15** |
| **Akchat Jain** | **16** |

**(Dr. Pratibha Kaladeep) (Dr.Mrs.A.M.Pujar)**

***Project Guide Head***

**Dept of Computer Sc. & Engg**

**(Dr. V.A. Athavale)**

**Principal**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**WALCHAND INSTITUTE OF TECHNOLOGY**

**SOLAPUR**

**(2024-2025)**

**Project Approval Sheet**

The Project Entitled

**TASK MANAGEMENT WEB APPLICATION**

**FOR SOFTWARE PROJECTS**

Submitted by

|  |  |
| --- | --- |
| **Piyush Agarwal** | **13** |
| **Venkat Dandi** | **14** |
| **Mahadev Dharmasale** | **15** |
| **Akchat Jain** | **16** |

“Is hereby approved in partial fulfilment for the degree of

Bachelor of Computer Science and Engineering”

**(Dr. Pratibha Kaladeep) (Dr. Mrs. A.M.Pujar)**

***Project Guide Head***

**Dept of Computer Sc. & Engg**

**(Dr.V.A.Athavale)**

***Principal***

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**WALCHAND INSTITUTE OF TECHNOLOGY**

**SOLAPUR - 413006**

**(2024-2025)**

**Acknowledgment**

At the outset, we would like to take this opportunity to express our deep gratitude to our guide **Project Guide** of CSE Department for his guidance and moral support throughout this successful completion of our project.

We heartily thank **Dr. Mrs. A.M.Pujar,** Head of CSE Dept for her moral support and promoting us through completion of our project.

We would also like to thank our Principal **Dr. V. A. Athavale** and all staff members for their whole hearted co-operation in completing this project.

**UNDERTAKING**

We solemnly declare that project work presented in the report titled **Project Title** is solely my project work with no significant contribution from any other person except project guide. Small contribution/help wherever taken has been duly acknowledged and that complete report has been written by the members of the project group.

We understand the zero tolerance policy of the WIT, Solapur towards plagiarism. Therefore we as authors of the above titled report declare that no portion of the report has been plagiarized and any material used as reference is properly referred / cited.

We undertake that if found guilty of any formal plagiarism in the above titled report even after award of the degree, WIT, Solapur and PAHSUS reserves the rights to withdraw/revoke the degree granted and that WIT, Solapur and the University has the right to publish our name on the website on which names of students are placed who submitted plagiarized report.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Exam Number | University PRN Number | Signature |
| Piyush Agarwal | 2102111476 | 2102111476 |  |
| Venkat Dandi | 2102111187 | 2102111187 |  |
| Mahadev Dharmasale | 2102111151 | 2102111151 |  |
| Akchat Jain | 2102111438 | 2102111438 |  |

Date: / /

**Abstract**

This project presents the development of a web-based application titled **Manage Mate**, designed to streamline task and project management while enhancing team collaboration and productivity. Inspired by tools like Bitrix24, the application was implemented with a focus on a three-tier hierarchical structure and role-based access control, enabling efficient oversight and task delegation across organizational roles.

The system supports three key user roles: Project Managers, Project Leaders, and Team Members. Core functionalities include role-specific dashboards, secure JWT-based authentication, task and project assignment, progress tracking, a point-based incentive system. The frontend was developed using React.js, styled with Tailwind CSS, while the backend uses Node.js with Express.js. MongoDB is used for database storage. This system lays a strong foundation for future enhancements like chat, Gantt charts, and advanced analytics.

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No.** | **Title** | | **Page No.** |
| **1** | **Introduction** | |  |
|  | 1.1 | Introduction |  |
|  | 1.2 | Problem Statement and Objective |  |
| **2** | **Background** | |  |
| **3** | **Proposed Solution** | |  |
|  | 3.1 | Solution |  |
|  | 3.2 | Advantages of proposed system |  |
| **4** | **Working Environment** | |  |
|  | 4.1 | Hardware Requirements |  |
|  | 4.2 | Software Requirements |  |
| **5** | **Methodology** | |  |
|  | 5.1 | System Architecture |  |
|  | 5.2 | Work Flow |  |
| **6** | **Flow Diagrams** | |  |
|  | 6.1 | Data Flow Diagram |  |
|  | 6.2 | Sequential UML Diagram |  |
| **7** | **Implementation** | |  |
|  | 7.1 | Code Snippet |  |
|  | 7.2 | Screenshots & results |  |
| **8**  **9** | **Testing Report**  **Project Cost (if required, based on project type)** | |  |
| **10** | **Conclusion** | |  |
| **11** | **Future Work** | |  |
| **12** | **References** | |  |

**Chapter 1**

**INTRODUCTION**

* 1. **Introduction :**

In today’s fast-paced and collaborative work environment, efficient task and project management remains critical for meeting organizational goals, enhancing team productivity, and maintaining streamlined operations. As teams expand and project scopes widen, the complexity of managing assignments, tracking progress, and ensuring accountability increases significantly. This project, *Manage Mate*, addresses these challenges by delivering a tailored web application designed to simplify and optimize project and task workflows through a role-based, hierarchical management structure.

*Manage Mate* draws inspiration from platforms like Bitrix24 but focuses on core project and task management features suitable for startups and growing teams. It defines three distinct user roles—**Project Manager**, **Project Leader**, and **Team Member**—each with clearly defined permissions. Project Managers oversee and assign projects to leaders, leaders divide projects into tasks for members, and members complete tasks while earning performance-based points. This structure fosters clarity, accountability, and smooth task delegation.

The system includes essential features such as project and task creation, progress tracking, and a performance leaderboard to motivate users. Role-specific dashboards, secure JWT-based authentication. Although initial ideas like real-time chat, meeting scheduling, and Gantt charts were considered, the final implementation focuses on delivering a robust and functional MVP (Minimum Viable Product) that supports task management, productivity tracking, and team coordination.

The application is built using **React.js** for the frontend, **Node.js** and **Express.js** for the backend, and **MongoDB** for database storage. JWT authentication is used to manage secure, role-based access. Ultimately, *Manage Mate* offers a scalable, flexible, and secure platform that simplifies team coordination and helps organizations improve execution and accountability in project delivery.

* 1. **Problem Statement and Objective**

In modern organizational settings, especially in small to mid-sized companies and start-ups, managing tasks and projects effectively is a common challenge. While many project management tools such as Bitrix24, Microsoft Teams, ClickUp, Jira, and Confluence offer a wide range of features, they often fall short when it comes to ease of use, flexibility, and adaptability to specific organizational structures.

The key issues identified with existing solutions are:

* Excessive feature bloat that makes the platform difficult to navigate for non-technical users.
* Lack of a streamlined, role-based access system that fits into a typical hierarchical team structure.
* Dependence on multiple platforms for communication, document sharing, and task management, leading to fragmentation and inefficiency.
* Limited customization for organizations that require lightweight solutions tailored to their workflow.
* Complex UI/UX in many tools that result in reduced adoption and engagement from team members.

These challenges highlight the need for a task management system that is intuitive, customizable, and capable of handling role-based access in a well-structured team environment.

**Objective**

The primary objective of this project is to develop **Manage Mate**, a web-based task management application that provides a simplified yet powerful solution for task and project management within hierarchical teams.

The key objectives of the system are as follows:

* To design and develop a task management platform that supports a **three-tier hierarchical structure** with clear **role-based access control (RBAC)** for Admin, Project Manager, Team Leader, and Team Member.
* To ensure a **user-friendly interface** that simplifies task delegation, project tracking, and daily communication among team members.
* To integrate essential features such as:
  + **Task creation, assignment, and deadline tracking**
  + **Real-time messaging/chat system**
  + **Project-wise categorization and progress tracking**
  + **Document sharing and file uploads**
  + **Performance tracking using a gamified points system**
* To implement **secure user authentication** using **JWT (JSON Web Tokens)** for safe and reliable login and access management.
* To support future integration of advanced modules such as **Gantt Charts, Analytics Dashboards, and Video Conferencing** capabilities.
* To provide an all-in-one platform that reduces dependency on multiple external tools, ensuring better collaboration and productivity.

In the realm of project and task management, numerous tools have been developed to enhance workflow efficiency, foster collaboration, and boost productivity. Industry-standard platforms like Bitrix24, ClickUp, and Microsoft Teams offer comprehensive feature sets, enabling teams to manage tasks, track project milestones, communicate in real-time, and collaborate on shared resources. However, despite their capabilities, these tools often pose usability challenges—especially for smaller teams or organizations with simpler workflows. Their extensive functionalities can lead to steep learning curves and increased setup complexity, which may hinder user adoption.

Other widely adopted project management solutions, such as Jira and Confluence, have earned strong reputations, particularly in the software development and documentation domains. Jira is well-regarded for its agile project planning, bug tracking, and sprint management, while Confluence excels in centralized documentation and team knowledge sharing. While powerful, these tools are often best suited for large teams or enterprises with advanced technical requirements. For non-technical teams or organizations with more streamlined needs, such platforms may prove overwhelming or underutilized, contributing to operational inefficiencies and reduced engagement.

Moreover, while integration between tools like Jira and Confluence can be advantageous, it often introduces complexity in managing multiple interfaces and understanding interdependencies. This fragmentation can hinder productivity and confuse users who require an all-in-one, cohesive platform that supports essential functions without overcomplicating the experience.

Recognizing these gaps, our project—**Manage Mate**—was developed as a lightweight, easy-to-use web application tailored for hierarchical team structures. It is designed to provide critical project and task management functionality with a simplified and accessible interface. Manage Mate maintains a clear separation of user roles—Project Manager, Project Leader, and Team Member—offering each user a distinct set of tools and permissions suited to their responsibilities. This structure ensures streamlined workflows and focused task delegation across the organization.

Key features adopted and refined from industry leaders include:-

* **Task Management**: Inspired by Jira, users can break down projects into tasks and assign them within a role-based hierarchy.
* **Task Tracking**: Like Jira’s robust tracking system, Manage Mate offers real-time task status updates (To-do, In Progress, Done) and progress visibility, implemented through a clean and intuitive dashboard.
* **Productivity Incentives**: A unique feature of Manage Mate is the point-based reward system for task completion, fostering motivation and transparency.

By addressing the limitations of existing platforms and focusing on user-friendly design, *Manage Mate* delivers an efficient solution for teams seeking a simpler, scalable, and practical project management tool. It bridges the gap between high-functionality enterprise tools and the needs of smaller, agile teams—allowing users to manage work effectively without unnecessary complexity.

**Chapter 2**

**Background**

**Chapter 3**

**PROPOSED SOLUTION**

**3.1 Solution**

The proposed project is a **web-based task and project management application** designed to enhance team collaboration, streamline task assignments, and provide robust tools for managing projects. The application will allow users to efficiently manage their tasks, communicate with teammates, track project progress, and generate reports—all within a user-friendly interface. This section outlines the detailed description of the project, including its core features, system architecture, and how it works.

**1. Overview of Features**

The web application will offer several key features aimed at simplifying task and project management:

**Core Features:**

1. **User Management:**
   * **Registration and Login**: Users can register for the platform and log in securely using **JWT authentication**.
   * **Role-Based Access Control (RBAC)**: The system supports multiple user roles (Admin, Manager, Team Leader, Team Member), each with distinct permissions for managing tasks, projects, and users.
2. **Task and Project Management:**
   * **Task Creation & Assignment**: Tasks can be created, assigned to team members, and tracked for progress.
   * **Project Creation & Management**: Managers and team leaders can create projects, define milestones, and allocate resources.
   * **Gantt Chart**: Visual representation of project timelines, task dependencies, and progress, allowing users to see the overall project schedule.
   * **Timeline View**: Projects include a timeline interface to help users visualize project phases and task durations.
3. **Communication Tools:**
   * **Real-Time Chat**: Users can communicate through a built-in real-time messaging feature, facilitating quick discussions within the team. Messages are instantly delivered and displayed using Socket.IO for seamless experience.
4. **Document Management:**
   * **File Sharing & Storage**: Users can upload, share, and access documents related to tasks and projects.
   * **Version Control**: Keeps track of file versions, ensuring that team members always access the most up-to-date documents.
5. **Reporting and Analytics:**
   * **Task & Project Reports**: Users can generate detailed reports on task completion, project status, and performance metrics.
   * **Dashboard Analytics**: Visual dashboards to display key performance indicators (KPIs), task progress, and team performance.
6. **Hierarchical Management:**
   * **Admin Panel**: Admins can monitor users and projects, and manage user roles and permissions.
   * **Project Manager & Team Leader Dashboards**: Role-based dashboards allow leaders to oversee projects and delegate tasks efficiently.
   * **Team Member Interface**: Members can view their assigned tasks, mark progress, and participate in team chats.

**2. System Architecture**

The architecture of the web application follows a **client-server model** where the frontend communicates with the backend via REST APIs. The system is divided into three primary components:

1. **Frontend (Client-Side)**:
   * Built using **React.js**, the frontend handles user interactions and communicates with the backend to fetch and display data.
   * **Bootstrap** is used to ensure responsive design and provide a consistent user experience across devices.
   * **Socket.IO** is used for real-time communication features, such as live chat and notifications.
2. **Backend (Server-Side)**:
   * The backend is built using **Node.js** and **Express.js**, providing a RESTful API to handle requests from the frontend.
   * **JWT (JSON Web Tokens)** is used for secure authentication and authorization.
   * The backend is responsible for managing business logic, handling task/project data, managing user roles, and ensuring security.
3. **Database (Data Storage)**:
   * The database is **MongoDB**, a NoSQL database, chosen for its flexibility and scalability.
   * It stores user data, task/project details, communication logs, and other necessary documents.
   * Data is stored in a structured, document-based format, making it easy to manage complex relationships (e.g., tasks within projects).

**3. Detailed Working**

**User Registration & Login:**

* Users register by providing their details (name, email, password).
* The system sends a confirmation email to verify the user's account.
* After successful registration, users log in, and the backend validates the credentials.
* Upon successful login, a **JWT token** is generated and sent to the client, allowing for stateless authentication on future requests.

**Role-Based Access Control (RBAC):**

* Depending on the role assigned (Admin, Manager, Team Leader, Team Member), users will be granted specific permissions:
  + **Admin**: Full access to manage users, roles, and view system performance.
  + **Manager**: Can oversee multiple teams, assign tasks, and generate reports.
  + **Team Leader**: Can assign tasks, monitor progress, and report project status.
  + **Team Member**: Can view and update their assigned tasks.

**Task and Project Management:**

* **Managers and Team Leaders** create new projects, define project milestones, and allocate tasks to team members.
* Tasks are created with specific deadlines, descriptions, and priorities. Team members can update the status of their tasks (e.g., Not Started, In Progress, Completed).
* The **Gantt Chart** feature helps visualize task dependencies, timelines, and milestones across a project.
* Real-time updates on tasks and project progress are reflected in the UI, allowing all team members to stay on the same page.

**Communication:**

* The **real-time chat** feature uses **Socket.IO** to allow team members to send instant messages. Messages are displayed in the chat interface without needing to reload the page.
* **Video conferencing** functionality is integrated directly into the platform, enabling remote teams to have meetings without leaving the app.

**Document Management:**

* Users can upload, download, and share files related to their tasks or projects.
* The application tracks versions of documents to ensure that the latest version is always accessible to the team.

**Reporting & Analytics:**

* Project managers and admins can generate detailed **reports** about the project's progress, task completion, team productivity, and other performance metrics.
* **Dashboards** display key metrics and visual graphs that provide an overview of the project's health and timelines.

**3.2 Advantages of the proposed system:**

The proposed system, **Manage Mate**, offers several advantages that address the limitations of existing task management platforms and bring added value to organizations, especially those operating in a structured, team-based environment.

**1. Hierarchical Role-Based Access**

* The system supports a **three-tier role hierarchy** (Admin, Project Manager, Team Leader, and Team Member), ensuring that users only access features and data relevant to their role.
* This minimizes confusion and enhances security by restricting unauthorized access.

**2. Centralized Task and Project Management**

* All task-related activities such as creation, assignment, tracking, and completion are centralized in one platform.
* Projects are organized clearly, making it easier to monitor progress and team productivity.

**3. Real-Time Collaboration**

* The integrated **chat and messaging system** enables instant communication among team members without needing external tools.
* This improves collaboration and reduces the need for frequent emails or meetings.

**4. User-Friendly Interface**

* Designed with **React.js and Tailwind CSS**, the system provides a clean and intuitive interface that requires minimal learning time, encouraging adoption by all team members.

**5. Secure Authentication**

* The system uses **JWT-based secure authentication**, ensuring that all user data and sessions are protected from unauthorized access.

**6. Enhanced Productivity with Performance Tracking**

* The system includes a **gamified points and progress tracker** that motivates users by recognizing their contributions.
* Visual indicators and dashboards help in monitoring both individual and team performance.

**7. Document and File Management**

* Users can easily **upload and share files** related to tasks and projects, ensuring all resources are organized and accessible from a single platform.

**8. Modular and Scalable Design**

* The application is designed to be **modular**, allowing new features (e.g., Gantt charts, analytics, video conferencing) to be integrated easily in future versions.
* Its scalable architecture ensures smooth performance as the number of users and tasks grows.

**9. Reduction in Tool Fragmentation**

* By combining task management, communication, file sharing, and performance tracking into one platform, the system **eliminates the need for multiple tools**, reducing costs and increasing efficiency.

**10. Better Decision Making**

* The clear visualization of project status, deadlines, and task ownership enables **faster and more informed decision-making** by managers and team leads.

**Chapter 4**

**WORKING ENVIRONMENT**

**4.1 Software Requirements:**

The proposed system requires the following software components for both development and deployment. These tools and technologies are chosen based on compatibility, performance, scalability, and ease of integration.

**1. Frontend Technologies**

* **React.js** – For building a dynamic and responsive user interface.
* **Tailwind CSS** – For utility-first, customizable, and responsive UI design.
* **Axios** – For making HTTP requests from the frontend to the backend.
* **React Router DOM** – For handling navigation and routing within the single-page application.

**2. Backend Technologies**

* **Node.js** – JavaScript runtime environment for executing backend code.
* **Express.js** – Web application framework for building RESTful APIs.
* **Multer** – For handling file uploads (e.g., profile pictures).
* **Jsonwebtoken (JWT)** – For implementing secure user authentication and session management.
* **bcrypt.js** – For hashing user passwords securely.

**3. Database**

* **MongoDB** – NoSQL database for storing user data, tasks, projects, messages, and role-based access control data.
* **Mongoose** – ODM (Object Data Modeling) library for MongoDB, used for schema creation and database queries.

**4. Development Environment**

* **Visual Studio Code (VS Code)** – Source code editor used for writing and editing project files.
* **Postman** – For API testing and debugging RESTful endpoints.
* **MongoDB Compass** – GUI for managing and visualizing MongoDB databases.

**5. Browser & OS Requirements**

* **Modern Web Browsers** – Google Chrome, Mozilla Firefox, Microsoft Edge (latest versions recommended).
* **Operating System** – Windows 10/11, macOS, or any Linux distribution (for both development and hosting).

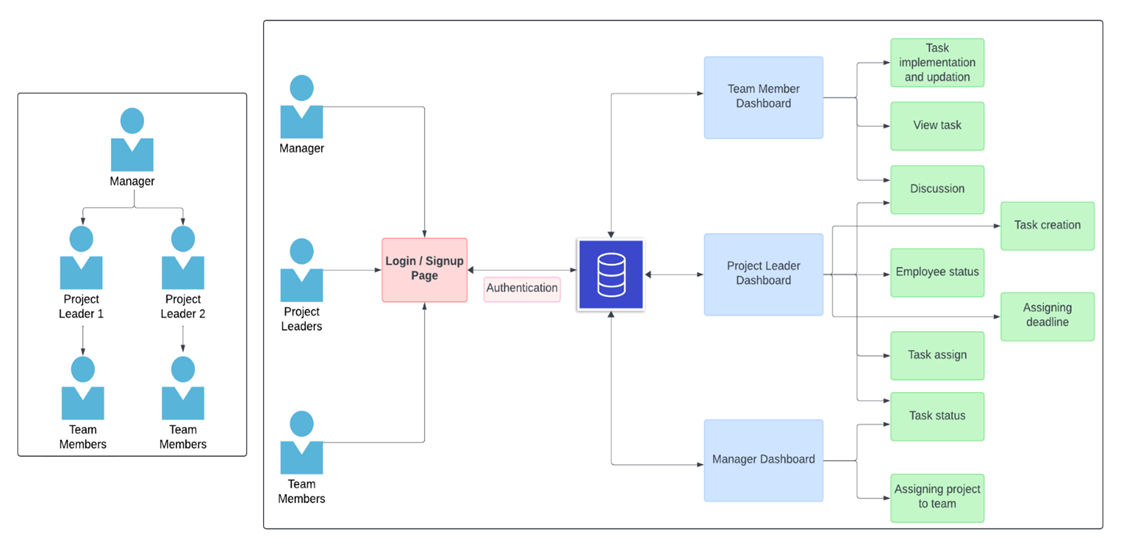
**6. Version Control & Collaboration**

* **Git** – Version control system for tracking changes in source code.
* **GitHub** – Remote repository hosting platform for collaborative development

**Chapter 5**

**METHODOLOGY**

**5.1 System Architecture**

**5.2 Work flow**

**Chapter 6**

**FLOW DIAGRAMS**

**6.1 Data flow diagrams**

**6.2 Sequential UML Diagrams**

**Chapter 7**

**IMPLEMENTATION**

**7.1Code Snippet**

**7.2 Screen shots& Results**

**Chapter 8**

**Testing Report**

**Chapter 9**

**Project Cost (if required, based on project type)**

**10. CONCLUSION**

The development of the *Manage Mate* Task Management Web Application has been successfully completed, delivering a fully functional and role-driven project management platform. The application provides a clean, responsive interface for managing tasks and projects, coupled with robust backend functionality that ensures secure authentication, efficient data handling, and smooth role-based access control.

Key features such as real-time chat using Socket.IO, timeline visualization, and user-specific dashboards have been effectively integrated. These components work together to streamline communication, improve team coordination, and enhance task visibility across all user roles—Admin, Project Manager, Team Leader, and Team Member.

By focusing on practical usability and scalable architecture, *Manage Mate* offers a simplified yet powerful alternative to traditional project management tools. The current version successfully meets the core requirements of collaborative task handling, hierarchical user interaction, and real-time updates, laying a strong foundation for future scalability.

Overall, the application presents a reliable, intuitive, and efficient solution for organizations seeking to manage projects and teams with greater clarity and control.

**11. FUTURE WORK**

**12. REFERENCES**

**Existing Model**

* ClickUp: [https://clickup.com/features/dashboards](https://www.canva.com/link?target=https%3A%2F%2Fclickup.com%2Ffeatures%2Fdashboards&design=DAGO1Vbatag&accessRole=owner&linkSource=document)
* Bitrix24: [https://bitrix24.in](https://www.canva.com/link?target=https%3A%2F%2Fwww.bitrix24.net%2Fcreate%2F%3Fuser_lang%3Den%26client_id%3Dsite.5602b8e8a41202.94213120%26ga%3DGA1.1.827600052.1724407445%26gclid%3DCjwKCAjwuMC2BhA7EiwAmJKRrFctHUqAgywZHEeEX6IGaeyLVTRqT-6GBt2-zYkxP4aDDrppUGg_wxoCrw8QAvD_BwE%26bx_utm_source%3Dgoogle%26bx_utm_medium%3Dcpc%26bx_utm_campaign%3D21442833010-161482363422%26bx_fbp%3Dfb.1.1724407445756.55450625026965662%26bx_user_agent%3DMozilla%252F5.0%2B%2528Windows%2BNT%2B10.0%253B%2BWin64%253B%2Bx64%2529%2BAppleWebKit%252F537.36%2B%2528KHTML%252C%2Blike%2BGecko%2529%2BChrome%252F128.0.0.0%2BSafari%252F537.36&design=DAGO1Vbatag&accessRole=owner&linkSource=document)
* Microsoft Teams: [https://teams.microsoft.com/v2/?  
  clientexperience=t2](https://www.canva.com/link?target=https%3A%2F%2Fteams.microsoft.com%2Fv2%2F%3Fclientexperience%3Dt2&design=DAGO1Vbatag&accessRole=owner&linkSource=document)
* Jira: [https://www.atlassian.com/software/jira](https://www.canva.com/link?target=https%3A%2F%2Fwww.atlassian.com%2Fsoftware%2Fjira%3F%26aceid%3D%26adposition%3D%26adgroup%3D143485223644%26campaign%3D18442427757%26creative%3D696980129686%26device%3Dc%26keyword%3Djira%26matchtype%3De%26network%3Dg%26placement%3D%26ds_kids%3Dp73345677068%26ds_e%3DGOOGLE%26ds_eid%3D700000001558501%26ds_e1%3DGOOGLE%26gad_source%3D1%26gclid%3DCj0KCQjw_sq2BhCUARIsAIVqmQtfpe10Lg9eX7C5NzbIgvnadQJ2ucXbW_iOPgEbDsbj7h_6d5wF5IYaAv9QEALw_wcB%26gclsrc%3Daw.ds&design=DAGO1Vbatag&accessRole=owner&linkSource=document)
* Confluence: [https://www.atlassian.com/software/confluence](https://www.canva.com/link?target=https%3A%2F%2Fwww.atlassian.com%2Fsoftware%2Fconfluence%3F%26aceid%3D%26adposition%3D%26adgroup%3D140527013959%26campaign%3D18336813584%26creative%3D656392809975%26device%3Dc%26keyword%3Dconfluence%26matchtype%3De%26network%3Dg%26placement%3D%26ds_kids%3Dp73000477014%26ds_e%3DGOOGLE%26ds_eid%3D700000001542923%26ds_e1%3DGOOGLE%26gad_source%3D1%26gclid%3DCj0KCQjw_sq2BhCUARIsAIVqmQtuT0szti-laZ46JF_9GNeG0Rmpp3rknpoGmbMw6hcSJWYcR1iwLP0aAhoCEALw_wcB%26gclsrc%3Daw.ds&design=DAGO1Vbatag&accessRole=owner&linkSource=document)

**Research Papers:**

* Almeida, J.R., Gini, R., Roberto, G. et al. TASKA: A modular task  
  management system to support health research studies. BMC Med  
  Inform Decis Mak 19, 121 (2019). [https://doi.org/10.1186/s12911-019-  
  0844-6](https://www.canva.com/link?target=https%3A%2F%2Fdoi.org%2F10.1186%2Fs12911-019-0844-6&design=DAGO1Vbatag&accessRole=owner&linkSource=document)
* Whittaker, S. (2005). Supporting Collaborative Task Management in  
  E-mail. Human–Computer Interaction, 20(1–2), 49–88.  
  [https://doi.org/10.1080/07370024.2005.9667361](https://www.canva.com/link?target=https%3A%2F%2Fdoi.org%2F10.1080%2F07370024.2005.9667361&design=DAGO1Vbatag&accessRole=owner&linkSource=document)